Learning From a Small Number of Training Examples by Exploiting Object Categories

Kobi Levi
School of Computer Science and Engineering
The Hebrew University of Jerusalem
91904 Jerusalem, Israel
kobilevi@cs.huji.ac.il

Michael Fink
Center for Neural Computation
The Hebrew University of Jerusalem
91904 Jerusalem, Israel
fink@huji.ac.il

Yair Weiss
School of Computer Science and Engineering
The Hebrew University of Jerusalem
91904 Jerusalem, Israel
yweiss@cs.huji.ac.il

Abstract

In the last few years, object detection techniques have progressed immensely. Impressive detection results have been achieved for many objects such as faces [11, 14, 9] and cars [11]. The robustness of these systems emerges from a training stage utilizing thousands of positive examples. One approach to enable learning from a small set of training examples is to find an efficient set of features that accurately represent the target object. Unfortunately, automatically selecting such a feature set is a difficult task in itself.

In this paper we present a novel feature selection method that is based on the notion of object categories. We assume that when learning to recognize a new object (like an apple) we also know a category it belongs to (fruit). We further assume that features that are useful for learning other objects in the same category (e.g. pear or orange) will also be useful for learning the novel object. This leads to a simple criterion for selecting features and building classifiers. We show that our method gives significant improvement in detection performance in challenging domains.

1 Introduction

Achieving human like object detection capabilities is one of the most challenging goals facing the computer vision community. The primary difficulty in achieving robust object detection emerges from the wide variety in the appearance of objects. In order to confront this variety, many recent object detection systems (e.g. [10, 11, 14]) use thousands of positive examples as a training set. Though using numerous examples enhances detection performance, we maintain several motivations for attempting to develop new detection methods that rely on small training sets.

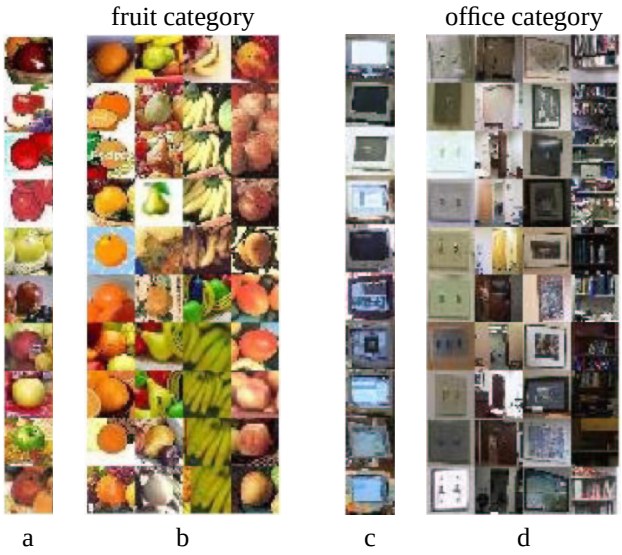

Figure 1: We demonstrate our approach on two categories: fruit (apple, orange, pear, banana and peach) and office objects (computer monitor, light switch, door, picture frame and bookshelf). Detection after training from a small sample (a - 10 apples, c - 10 monitors), was significantly improved by prioratizing features with high discriminative value on the remaining objects in the category (b,d).

Our first motivation stems from the human ability to detect new objects after a short exposure to very few examples [5]. In addition training examples are often expensive to acquire or otherwise scarce. Collecting 5,000 examples

of upright faces might be a matter of a few days work using an Internet search engine, but obtaining a large number of annotated images of other objects like apples or chairs is a substantially more demanding task.

As the object detection research progresses, more and more feature sets are proposed for different object detection tasks [9, 14, 1, 8]. One might assume that using an inclusive pool of all feature sets proposed in the literature might be instrumental in creating a generic object detection scheme. However, adopting such a large feature pool leads to an overfitting problem when learning from few training examples. We therefore remain with the option to either manually select which feature set should be used for each specific object or characterize a generic mechanism that automatically selects the relevant feature set.

Defining a relevant feature set is often done implicitly by the researchers' biased selection of the candidate feature set. Yet, this selection might be difficult and counterintuitive. For example Haar-features have been surprisingly effective in characterizing upright frontal faces [14], but when applying these same features for profile face detection a substantial drop of performance is observed [9]. Our underlying assumption is that while defining an efficient feature set might be nontrivial; characterizing a perceptual category by stating a visual resemblance might be more natural.

Applying any traditional learning algorithm to each of the objects in a visual category independently would yield poor detection results due to combined effect of a small available training sample and a substantial candidate feature set.

Rather than relying on many examples of a single object, our approach assumes that a few examples of many objects are available and that we have some capability of segregating the world of objects into different visual categories[1]. It should be emphasized that we focus on utilizing visual categories, defined by perceptual commonalities and not on semantic categories, which might be defined by function or abstract knowledge (like genetic resemblance). We show that using the information common between several objects in the perceptual category can significantly improve the detection results of any novel object belonging to the same category.

1.1 Previous works

The human perceptual system possesses the unparalleled capability of learning to detect a new object category from very few examples. This capacity might be a necessity in designing a perceptual system capable of recognizing 30,000 objects [2]. In the last decades the computer vision community has struggled to achieve robust object detection capabilities [11, 14]. These methods notoriously require thousands of training examples. Lately, comparable object detection performance has been achieved with as few as 250 training examples [9].

As the single object detection algorithms have matured, several researchers attempted to detect multiple objects in an image (e.g. [8]). Nevertheless, the required number of training examples for these multi class object detectors has not significantly decreased as a function of the multiple learning tasks. One exception can be found in an attempt to learn priors on object appearance and shape configurations by training many object classifiers in a Bayesian framework [3]. Recently it has been demonstrated that robust detection capabilities of multiple objects might be achieved by selecting a small set of common features [12]. Given information on such a set of features, we will demonstrate that training from a very small sample might be feasible.

2 Approach

Assume we have a learning algorithm aimed at generating an object detector. The input of this algorithm is a set of positive and negative examples of a single category and its output is a binary classifier $\mathcal{H}(x)$. Given any candidate image sub-window x, $\mathcal{H}(x)$ might classify it either as an example of the target object or otherwise reject the sub-window. The classifier's output is determined by some attributes of x commonly termed, features. In Section 3 we describe a broad set of features which capture many of the objects properties such as shape and color. Although the examples are of a relatively small size (e.g. 24x24 pixels) this set is quite extensive (including over $1,000,000$ features). Due to detection efficiency constraints, we would like $\mathcal{H}(x)$ to rely only on a small fraction of the immense candidate feature set. Thus, the detection learning algorithm must include a mechanism for feature selection.

When one examines the substantial feature set, it is obvious that several features generalize better than others for many objects. Thus, for example, features that are very small (e.g. relying on a single pixel) are expected to show poor generalization results. However, when trained on a small set of examples, such incidental features can occasionally exhibit high discriminative performance between the positive and negative examples of a restricted training set. It is observed that features that generalize well are often characterized as being of intermediate complexity [13].

In addition to this basic superiority that some features have over others, we wish to make use of the fact that different channels of information characterize different categories of objects. This implies that if a specific type of features (e.g. color features), have shown good performance

[1] We use the term category to describe a group of objects with some common ground e.g. the category fruit includes the objects orange, pair etc.

over objects in a certain visual category, this type of features will probably generalize well in other objects in the same visual category.

Hence, our algorithm includes two stages:

1. Estimate a category related error for each feature. Given the category j, the output of this stage is a group of expected errors C_i^j ($0 \leq C_i^j \leq 1$) where i stands for the feature i.

2. Incorporate these category-related errors into our object detection learning scheme in order to create a more robust and accurate classifier.

The following section describes how we calculate these category related errors and how we embed them into our object detection learning scheme.

2.1 Learning category related errors

When considering the detection tasks of apples and oranges, it is important to notice that though colors are important in both tasks, it does not mean that all fruit have the same color. Hence, when we learn the category related errors, it is not reasonable to assign a unique error for each feature (e.g. color is red) but rather to a group of features, which are grouped together based on there characteristics such as type and size.

We therefore divide the immense feature set, $\mathcal{F}$, into K bins where each bin includes features from a specific type and a specific size. Thus, $K = |types| * |sizes|$.

For each bin k we will calculate its category related error B_k^j where j stands for the category. The category related error of each feature is then determined by its bin i.e., $C_i^j = B_k^j$ if feature i belongs to the bin k.

We propose a simple method to calculate B_k^j. Given an object m that belongs to the category j, we define $Error_i^m$ to be the classification error that the feature F_i can achieve on the training examples of object m. I.e,

$$error_i^m = \frac{\#FP}{\#Falses} + \frac{\#FN}{\#Positives} \quad (1)$$

Where $\#FP$ is the number of false positives and $\#FN$ is the number of false negatives.

Notice that $0 \leq error_i^m \leq 1$ and $error_i^m = 0$ iff F_i correctly classifies all examples. Notice also that $error_i$ does not depend on the ratio between the number of positive and negative examples. This is required since in object detection settings the number of negative examples significantly exceeds the number of positive examples in the training set.

We then set B_k^j to be the average error over all the features in bin k and over all objects in category j, i.e,

$$B_k^j = mean_{m \in category_j,\ i \in bin_k}(error_i^m) \quad (2)$$

2.2 Embedding category related errors within a boosting framework

The main idea proposed in this paper is that by using category related information regarding the features, over-fitting might be prevented. Thus, we avoid cases where a feature selection procedure might choose feature F which classifies the small training set well but later demonstrates poor generalization capabilities.

Though many learning algorithms can be modified in order incorporate category related information, we adopted AdaBoost [6]. The AdaBoost algorithm is widely used for object detection (for example [9, 14, 12]). As we will show, we can apply our concept of using prior knowledge over the features while remaining within the robust framework of AdaBoost.

Recall that AdaBoost requires that for any weight distribution D over the examples the weak learner will be able to return a weak hypothesis $h_i(x) \in \{1, -1\}$ which performs better than chance i.e. its weighted error on the training examples, Err_i is less than 0.5.

Often, the weak hypothesis $h_i(x)$ consists of a single feature $F_i(x)$ and a threshold T where

$$h_i(x) = \begin{cases} 1 & \text{if } F_i(x) \geq T_i \\ -1 & \text{otherwise} \end{cases} \quad (3)$$

When using AdaBoost as a feature selection mechanism, it is common that the weak learner chooses the hypothesis $h_i(x)$ with minimal error Err_i over the weighted distribution D of the training examples. Thus, on iteration t of the boosting process, the weak learner is usually guided to choose the weak hypothesis according to the following rule:

$$h^t(x) = argmin_{h_i(x)}(Err_i) \quad (4)$$

However, as we mentioned above, AdaBoost does not require that the weak learner will return the weak hypothesis $h_i(x)$ with the minimal training error Err_i but rater that it returns any weak hypothesis $h_j(x)$ with an error smaller than 0.5. Therefore, as long as we maintain the weak learner criterion we can change the conventional weak learner selection such that it will consider also category related errors estimated at the previous stage:

$$h^t(x) = argmin_{h_i(x), Err_i < 0.5}(\gamma * Err_i + (1-\gamma) * C_i^j) \quad (5)$$

Where γ determine the relative emphasis of the category related error ($0 \leq \gamma \leq 1$).

We summarize our approach as an attempt to achieve a more robust estimation of each feature's error by making use of the feature's discriminative capacity on other objects within the same visual category. Our experiments will show

that by doing so features that incidentally discriminate between the positive and negative examples in the small sample and that will lead to overfitting effects, will not be preferred over features that have a well established discriminative capacity for the remaining objects within the category.

3 Features

The multiplicity of features suggested for different object detection tasks demonstrates that no one single type of features might suffice for all possible detectors. Therefore, we created a pool of features that characterizes many aspects of the object like shape, color and contour.

Though our model does not assume any efficiency constraints applying to the features, due to practical reasons, we limited ourselves to features that can be quickly calculated during the detection phase. Thus, all the proposed features can be efficiently calculated so that the resulting classifier will have real-time performance. Specifically, after a preprocessing phase which takes $O(n)$ steps with a small constant (where n is the number of pixels in the image), calculating the value of each feature on any scale requires only $O(1)$ basic operations. We will now turn to describe in detail, the set of proposed features.

3.1 Haar-like Features

In [14] Viola and Jones present a set of features that measure the contrast between two to four neighboring areas in the sub-window. Suppose that R_1 and R_2 are two neighboring rectangles in the image and that $val(p)$ is the gray level of the pixel p then $F_{haar}(R_1, R_2)$ is defined to be:

$$F_{haar}(R_1, R_2) = \sum_{p \in R_1} (val(p)) - \sum_{p \in R_2} (val(p)) \qquad (6)$$

It is easy to infer the computation of this feature for the case of three or four neighbouring areas. It has been shown that these features can be calculated efficiently with only a small constant number of lookup table operations using the *Integral Image* data structure. See figure 2 for an illustration of these features.

3.2 Edge Orientation Features

While the Haar-like features perform well on some tasks such as face detection, they fail when the contrast relations do not tend to be constant. Therefore, Levi and Weiss [9] proposed a new set of features which captures the relations between two edge orientations within a rectangle.

These features are calculate by first performing edge detection using Sobel masks. Then the edges are grouped into four representative orientations. Let $E_k(R)$ be the sum of edges from the group k in the rectangle R. Then $F_{eoh}(R, K_1, K_2)$ is defined to be:

$$F_{eoh}(R, K_1, K_2) = \frac{E_{k_1}(R) + \epsilon}{E_{k_2}(R) + \epsilon} \qquad (7)$$

Levi and Weiss show that these features can be calculated efficiently using the *Integral Image*. These features are superior to the previous Haar-like features when the detection is based mainly on the outer contour of the object.

3.3 Dominant Orientation Features

Several objects are characterized by a dominant edge orientation in a specific area rather than the ratio between two different orientations. Therefore Levi and Weiss defined an additional set of features, which measure the ratio between a single orientation K and the others remaining orientations, i.e.

$$F_{de}(R, K) = \frac{E_k(R) + \epsilon}{\sum_i E_i(R) + \epsilon} \qquad (8)$$

3.4 Color Features

While many objects can be detected solely based on their shape, it is clear that color plays a significant role in the detection process of certain objects. Therefore, we created a set of features that capture information from the color space. Assuming that we work in the RGB color space, it is well known that this color representation is not invariant to illumination. In order to overcome this problem, Gevers and Smeulders [7] proposed the $l1, l2, l3$ color model.

$$l1 = \frac{(R-G)^2}{Z}, \quad l2 = \frac{(R-B)^2}{Z}, \quad l3 = \frac{(G-B)^2}{Z} \qquad (9)$$

Where

$$Z = (R-G)^2 + (R-B)^2 + (G-B)^2 \qquad (10)$$

Based on the $l1, l2, l3$ model, we define a new set of features, F_{color}, that measure the intensity of one of the $l1, l2, l3$ channels over the rectangle R.

$$F_{color}(R, l_x) = \sum_{p \in R} val(l_x(p)) \qquad (11)$$

4 Implementation

In this section we will briefly describe the cascade data structure [14] which we use in this paper.

In order to detect an object in an image we need to examine each possible sub-window and determine whether it

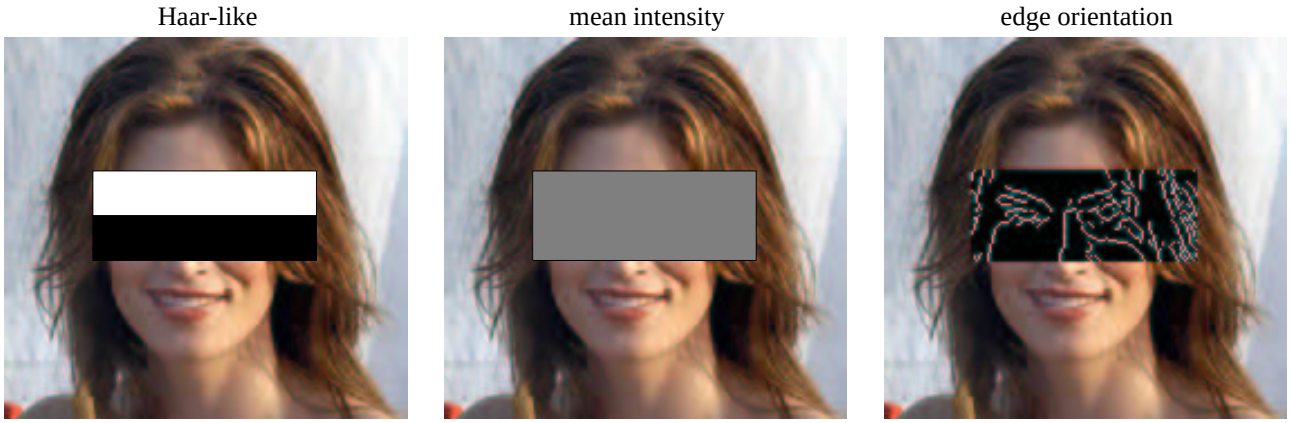

Figure 2: Three of the feature types implemented in this paper.

contains a face or not. In a regular image of 320*240 pixels there are over 500,000 sub-windows. In order to reduce the total running time of the system, we need to radically limit the average time that the system spends on processing each sub-window. For this purpose, Viola and Jones [14] suggested using a cascade of classifiers. The idea of a cascade is based on the observation that we need very few features to create a classifier that accepts almost all (more than 99%) positive examples while rejecting non-negligible amount (20 - 50%) of the false examples. Linking many such classifiers in a sequence creates a cascade that separates positive and negative examples in a robust manner. This is done with a very low cost per sub-window due to the fact that most non-face sub-windows are rejected in the early classifiers of the cascade. In order to train each stage of the cascade, we use the discrete version of Adaboost [6] to select features and determine their weights. The classifier at stage t of the cascade is:

$$H_t(x) = sign(\sum_{i=1}^{n} \alpha_i h_i(x)) \qquad (12)$$

where $h_i(x)$ is a weak hypothesis and α_i is its weight.

5 Experiments

5.1 Dataset

In order to test our method, we collected examples of objects from two categories, fruits and in-door office objects. Each category includes 5 objects. The fruit category includes the objects apple, orange, pear, banana, and peach. The indoor office category includes the objects computer monitor, light switch, door, picture and book shelf. The examples of fruit were taken from the Internet (using Google queries with the object name). We used the Caltech office database [4] to extract examples of common office objects. The entire set of positive examples is shown in figure 1.

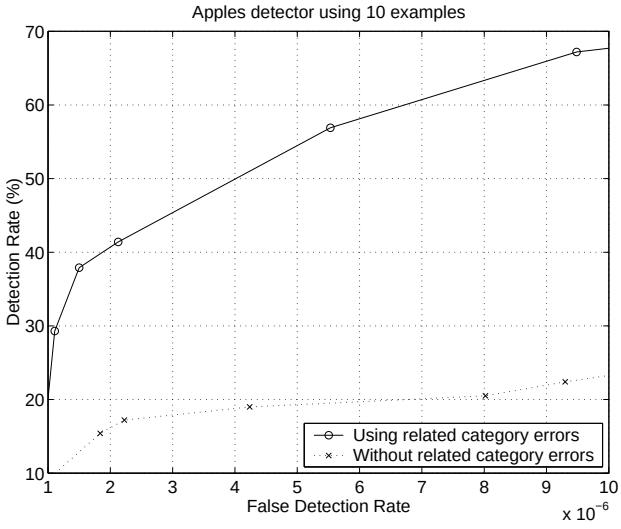

Figure 3: Results of the apple detector.

All examples were rescaled to 24x24 but no other preprocessing (such as alignment, normalization etc.) has been performed. We made use of a set of $\sim$ 500 images, which do not include any of the objects, so that by randomly cropping these images, we were able to extract the required number of negative examples (5000 false examples).

5.2 Testing

We define our goal as learning to detect a target object (e.g. computer monitors or apples) from 10 positive examples (along with 5000 randomly selected false examples).

We begin testing our two stage algorithm by estimating the category related error of each feature on the remaining

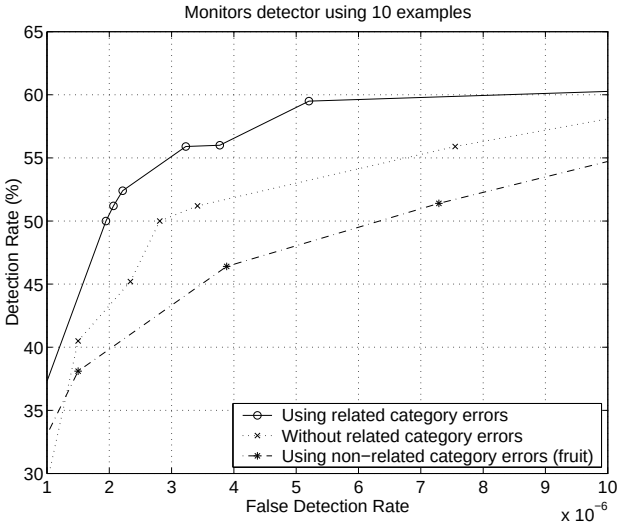

Figure 4: Results of the monitor detector.

category objects (see figure 1).

We then compare three training procedures:

- We first followed the naive object detection algorithm which does not use the category related errors. As described above, our training set included 10 positive examples for each object. We refer to the results of this procedure as a baseline for later comparison.

- In the second training procedure we incorporate the category specific errors while training the target object detector. In this experiment we set the value of γ to 0.5 (see equation 5), i.e. we equally weight the category related errors and the current training error of the target object.

- One may argue that using any similar constraint on the feature space, might improve detection results due to model regularization effect. In order to eliminate this possibility, we perform a third training procedure. In this setup, rather than using the category related errors we imposed errors calculated for another category. In other words, we used the fruit category errors instead of the office category errors while training a monitor detector.

In the following section we compare the results of these three training procedures on a predefined test set composed of a 100 images containing the target objects.

5.3 Results

We demonstrate the effect of using category related errors on two objects, one from each category. Figure 3 shows the results of the apple detector, whereas figure 4 shows the results of the monitor detector.

The results of the apple detector show that incorporating category related errors can have a significant effect in improving the detection rates. Given a false positive rate of $2 * 10^{-6}$ the detection rates improve from 17.2% to 41.4% by using the fruit category related errors. Given a false positive rate of $1*10^{-5}$ using these errors improves the detection rates from 22.4% to 67.2%.

In a similar fashion, the results of the monitor detector demonstrate the advantage of using category related errors. While a naive detector achieves a detection rate of 55.9% with 1521 false positives ($\sim 7.5 * 10^{-6}$), a detector which uses the category related errors achieves the same detection rate with only 650 false positives ($\sim 3.3 * 10^{-6}$).

Finally, we examine whether the improvement in the detection rates results from utilizing the category related errors or just from the constraining the feature space. To this end we trained an object detector from one category (monitors) using the category related errors derived from the other category (fruit). As illustrated in figure 4 the detection rates of a detector trained using non-related category errors were substantially lower than the naive detector. This clearly shows that the improvement we achieved is due to the category specific information rather than due to just constraining the substantial feature space.

6 Discussion

One of the major challenges facing object detection research is developing algorithms that will scale up to recognition of thousands of objects. In this paper we have focused on one aspect of scaling up - developing recognition algorithms that require a small number of training examples. The basic insight behind our method is that the world of objects is not a uniform mixture, but rather consists of discrete object categories. By utilizing this additional source of information, our proposed method can achieve significant improvement in detection ability with a small number of training examples.

In our current method, we have assumed that we know for each object to which perceptual category it belongs. We are currently working on relaxing this assumption as well as investigating the use of overlapping object categories.

References

[1] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, pages 831–837, 2000.

[2] Irving Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 1987.

[3] Li Fei-Fei, Rob Fergus, and Pietro Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings.*, pages 1134–1141, 1998.

[4] Michael Fink. The full images for natural knowledge caltech office db., 2003.

[5] Michael Fink and Kobi Levi. Reusable perceptual features enable efficient detection learning, 2004. Submitted to Perception 2004.

[6] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[7] Theo Gevers and Arnold W. M. Smeulders. Color based object recognition. In *ICIAP (1)*, pages 319–326, 1997.

[8] William Freeman Kevin Murphy, Antonio Torralba. Using the forest to see the trees: A graphical model relating features, objects and scenes. In *NIPS*, 2003.

[9] Kobi Levi and Yair Weiss. Learning object detection from a small number of examples: the importance of good features, 2004. To appear in CVPR 2004.

[10] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[11] H. Schneiderman and T. Kanade. A statistical approach to 3d object detection applied to faces and cars. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 746–751, June 2000.

[12] Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Sharing visual features for multiclass and multiview object detection, 2004. To appear in CVPR 2004.

[13] Shimon Ullman, Michel Vidal-Naquet, and Erez Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5(7):682 – 687, July 2002.

[14] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.